

# 제 24장 클로저

## 렉시컬 스코프

JS 엔진은 함수를 어디서 호출했는지가 아닌 어디에 정의했는지에 따라 상위스코프를 결정  
이를 **렉시컬스코프**라 칭한다

## 함수객체의 내부슬롯

함수는 자신의 내부슬롯 `[[Environment]]` 에 자신이 정의된 환경 즉 상위스코프의 참조를  
저장한다

내부슬롯에 저장된 함수는 상위스코프 이다

상위스코프는 자신이 존재하는 한 기억(참조)한다.

가비지 컬렉터는 누군가 참조하고 있는 메모리 공간을 함부로 해제하지 않는다

고로 저장된 상위스코프는 함수가 존재하는 한 유지된다.

## 클로저

외부함수보다 중첩함수가 더 오래 유지되는경우 중첩함수는 이미 생명주기가 종료한 외부함  
수의 변수를 참조할 수 있다.

이러한 중첩함수를 **클로저**라 칭한다

## 클로저의 활용

클로저는 상태를 안전하게 변경하고 유지하기 위해 사용된다.

1. 상태를 안전하게 은닉
2. 특정함수에게만 상태변경을 허용

```
// 카운트 상태 변경 함수
const increase = (function () {
  // 카운트 상태 변수
  let num = 0;

  // 클로저
  return function () {
    // 카운트 상태를 1만큼 증가 시킨다.
    return ++num;
  };
})();

console.log(increase()); // 1
console.log(increase()); // 2
console.log(increase()); // 3
```